

An Ordering (Enumerative) Algorithm for Nonlinear 0-1 Programming

BÉLA VIZVÁRI¹ and FATİH YILMAZ²

¹*Department of Industrial Engineering, Bilkent University, Ankara; current address: Rutgers University RUTCOR, P.O. Box 5062, NJ 08903-5062, U.S.A., vizvari@rutcor.rutgers.edu;* ²*Dept. of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey, yilmaz@trbilun.bitnet*

(Received: 12 July 1993; accepted: 11 February 1994)

Abstract. In this paper, a new algorithm to solve a general 0–1 programming problem with linear objective function is developed. Computational experiences are carried out on problems where the constraints are inequalities on polynomials. The solution of the original problem is equivalent with the solution of a sequence of set packing problems with special constraint sets. The solution of these set packing problems is equivalent with the ordering of the binary vectors according to their objective function value. An algorithm is developed to generate this order in a dynamic way. The main tool of the algorithm is a tree which represents the desired order of the generated binary vectors. The method can be applied to the multi-knapsack type nonlinear 0–1 programming problem. Large problems of this type up to 500 variables have been solved.

Key words: Nonlinear integer programming, enumeration, lexicographic order, global optimum.

1. Introduction

The general 0–1 programming problem with linear objective function can be stated as:

$$\begin{aligned} \max \quad & cx \\ x \in S \subseteq & \{0, 1\}^n \end{aligned} \quad (\text{GBP})$$

where $c \in N_+^n$ and without loss of generality it is assumed in the whole paper that

$$c_1 \geq c_2 \geq \dots \geq c_n > 0. \quad (1)$$

It is well-known that (GBP) is a hard problem to solve if S has no special properties. A new algorithm to solve (GBP) is presented in this paper. The algorithm is tested on problems where the set S is defined with polynomial inequalities. Experiences show that the method is competitive, i.e. problems have been solved with at least the same size as those reported in [1] and [5]. The method can be applied to the multi-knapsack type nonlinear 0–1 programming problem which is the following

$$\begin{aligned} \max \quad & \sum_{i=1}^p c_i \prod_{j \in S_i} x_j \\ \sum_{k=1}^{n_i} a_{ik} \prod_{j \in S_{ik}} x_j \leq & b_i \quad i = 1, \dots, m \\ x_j \in \{0, 1\} \quad & j = 1, \dots, n, \end{aligned} \quad (\text{MKTNL})$$

where it is assumed that all coefficients are positive and $S_i, S_{ik} \subseteq \{1, \dots, n\}$. Large problems of this type have been solved exactly by the method described in Section 4. [5] has collected a lot of applications of nonlinear 0–1 programming. In many cases the underlying mathematical problem is (GBP). In some applications, e.g., in testing vision loss discussed in [6], this problem is (MKTNL).

The new method is an enumerative algorithm. Enumeration is a well-known technique to solve NP-hard problems. This type of algorithm can be found, e.g., in [2], [4], [7], [12], [14], [15] and [16].

The organization of the paper is as follows. Section 2 describes a solution scheme, which was applied earlier in several papers. A basic observation is presented in the same section. The tree representation of the ordering of binary vectors, which is the main algorithmic tool, is discussed in Section 3. The application of the algorithm to the multi-knapsack type nonlinear 0–1 programming problem can be found in Section 5. Finally the last section discusses the computation experiences.

2. Solution of GBP via a Sequence of Set Packing Problems

In [4] and [7], the following scheme is applied to solve Problem (GBP).

If $e \in S$, then it is optimal where e is the all 1 vector. If $e \notin S$, then the optimal solution of (GBP), if it exists, must satisfy the following inequality

$$\sum_{j=1}^n x_j \leq n - 1.$$

Let us consider the following problem :

$$\begin{aligned} Z_2 = \max \quad & cx \\ \text{s.t.} \quad & \sum_{j=1}^n x_j \leq n - 1 \\ & x \in \{0, 1\}^n \end{aligned} \quad (2)$$

and assume that optimal solution of (2) is β_2 . It can be seen that

$$\beta_{2j} = \begin{cases} 1 & \text{if } j \in \{1, \dots, n-1\} \\ 0 & \text{if } j = n. \end{cases}$$

Then, it is obvious that if $\beta_2 \in S$, so it is an optimal solution of (GBP). Now assume that $\beta_2 \notin S$, it follows that the optimal solution of original problem must satisfy

$$\sum_{j=1}^n x_j \leq n - 1$$

$$\beta_2 x \leq \beta_2 e - 1$$

$$x \in \{0, 1\}^n.$$